UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/801,792	03/16/2004	Michael A. Holmes	M. HOLMES 3-6	1925

47396 7590 10/17/2007

HITT GAINES, PC
LSI Corporation
PO BOX 832570
RICHARDSON, TX 75083

EXAMINER	
QIAN, SONGWEI	
ART UNIT	**PAPER NUMBER**
2191	

NOTIFICATION DATE	DELIVERY MODE
10/17/2007	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

docket@hittgaines.com

Office Action Summary	Application No.	Applicant(s)	
	10/801,792	HOLMES ET AL.	
	Examiner	Art Unit	
	Songwei Qian	2191	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *16 August 2007*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-20* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-20* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. The application claims benefit of 06/488,168 filed on 07/18/2003.

2. Claims 1-20 are pending in this application.

3. Claims 1 and 15 are amended by Applicant on 08/16/2007.

4. Claims 1-20 are presented for examination.

Claim Rejections - 35 USC § 101

5. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

6. Claims 1-7 and 15-20 are rejected under 35 U.S.C. 101 as the claimed invention is directed to non-statutory subject matter.

7. In claims 1-7 and 15-20, a "computer-implemented system" is being recited; however, it appears that the system would reasonably be interpreted by one of ordinary skill in the art as software, per se. The only three elements positively recited as part of the system are the "graph generator", "graph converter", and "description generator". It appears that these elements would reasonably be interpreted as representative of the software. As such, it believed that the system of claims 1-7 and 15-20 is reasonably interpreted as software, per se.

Claim Rejections - 35 USC § 103

8. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

9. Claims 1-2, 4-9, and 11-14 are rejected under 35 U.S.C. 103(a) as being unpatentable over Gregory et al. (US Pat. # 5,937,190), hereinafter "Gregory" in view of Nelson et al. (US Pat. # 5,568,644), hereinafter "Nelson", and further in view of Oualline (Steve Oualline, Practical C Programming, 3rd Edition, O'Reilly, August 1997).

10. As for claim 1, Gregory discloses:

A computer-implemented system for automatically generating a hierarchical register consolidation structure (A computer-aided circuit analysis tool, Col. 56, lines 52-54), comprising:

a graph generator that parses a High-level Design Language (HDL) file to generate an intermediate graph (Figures 4 and 9, and Col. 27, line 61; note that the tree in Figure 9 is considered as an intermediate graph);

a graph converter, associated with said graph generator, that selectively adds virtual elements and nodes to said intermediate graph (4135, Figure 7) to transform said

intermediate graph into a mathematical tree (Figures 7 and 12; the tree in Figure 12 is considered as a mathematical tree);

But Gregory does not explicitly disclose:

an intermediate graph containing definitions of microprocessor-accessible registers, node interrelationships and summary bits and masks associated with alarm registers;

a description generator, associated with said graph converter, that employs said mathematical tree to generate a static tree description in a programming language suitable for use by a device-independent condition management structure.

However, Nelson discloses:

an intermediate graph containing definitions of microprocessor-accessible registers, node interrelationships and summary bits and masks associated with alarm registers (Fig. 1A and Col. 3, lines 9-15 and lines 31-44; note that a certain register is a alarm register and an alarm register is a computer device that store summary bits and masks).

It would have been obvious to one of ordinary skill in the art at the time of invention was made to combine the teachings of Gregory with the teachings of Nelson by making an intermediate graph to contain definitions of microprocessor-accessible registers,

node interrelationships and summary bits and masks associated with alarm registers in order to determine the interrupting device (Nelson, Col. 1, line 51).

But neither Gregory nor Nelson explicitly disclose:

a description generator, associated with said graph converter, that employs said mathematical tree to generate a static tree description in a programming language suitable for use by a device-independent condition management structure.

However, Oualline discloses:

a description generator, associated with said graph converter, that employs said mathematical tree to generate a static tree description in a programming language suitable for use by a device-independent condition management structure (Chapter 17, Section 17.10: Data Structures for a Chess Program, Page 1 of 2).

It would have been obvious to one of ordinary skill in the art at the time of invention was made to combine the teachings of Gregory and Nelson with the teachings of Oualline by employing said mathematical tree to generate a static tree description in a programming language suitable for use by a device-independent condition management structure in order to organize and process information (Oualline, Chapter 1, Page 1 of 4, 1st paragraph).

11. As for claim 8, Gregory discloses:

A method of automatically generating a hierarchical register consolidation structure (A computer-aided circuit analysis tool, Col. 56, lines 52-54), comprising:

parsing a High-level Design Language (HDL) file to generate an intermediate graph (Figures 4 and 9, and Col. 27, line 61; note that the tree in Figure 9 is considered as an intermediate graph);

selectively adding virtual elements and nodes to said intermediate graph (4135, Figure 7) to transform said intermediate graph into a mathematical tree (Figures 7 and 12; the tree in Figure 12 is considered as a mathematical tree);

But Gregory does not explicitly disclose:

an intermediate graph containing definitions of microprocessor-accessible registers, node interrelationships and summary bits and masks associated with alarm registers;

employing said mathematical tree to generate a static tree description in a programming language suitable for use by a device-independent condition management structure.

However, Nelson discloses:

an intermediate graph containing definitions of microprocessor-accessible registers, node interrelationships and summary bits and masks associated with alarm registers (Fig. 1A and Col. 3, lines 9-15 and lines 31-44; note that a certain register is a alarm register and an alarm register is a computer device that store summary bits and

masks).

It would have been obvious to one of ordinary skill in the art at the time of invention was made to combine the teachings of Gregory with the teachings of Nelson by making an intermediate graph to contain definitions of microprocessor-accessible registers, node interrelationships and summary bits and masks associated with alarm registers in order to determine the interrupting device (Nelson, Col. 1, line 51).

But neither Gregory nor Nelson explicitly disclose:

employing said mathematical tree to generate a static tree description in a programming language suitable for use by a device-independent condition management structure.

However, Oualline discloses:

employing said mathematical tree to generate a static tree description in a programming language suitable for use by a device-independent condition management structure (Chapter 17, Section 17.10: Data Structures for a Chess Program, Page 1 of 2).

It would have been obvious to one of ordinary skill in the art at the time of invention was made to combine the teachings of Gregory and Nelson with the teachings of Oualline by employing said mathematical tree to generate a static tree description in a

programming language suitable for use by a device-independent condition management structure in order to organize and process information (Oualline, Chapter 1, Page 1 of 4, 1st paragraph).

12. As for claims 2 and 9, the claims are rejected for the same reasons as claims 1 and 8 above. In addition, Nelson discloses:

said intermediate graph further contains bit offsets associated with said alarm registers (Col. 3, lines 31-44; note that a certain register is a alarm register and an alarm register is a computer device that store bit offsets).

13. As for claims 4 and 11, the claims are rejected for the same reasons as claims 1 and 8 above. In addition, Oualline discloses:

said programming language is C (Chapter 17, Section 17.10: Data Structures for a Chess Program, Page 1 of 2).

14. As for claims 5 and 12, Gregory discloses:

said HDL file is produced by a hardware description tool (Col. 5, lines 19-22; note that VHDL is a hardware description tool).

15. As for claims 6 and 13, the claims are rejected for the same reasons as claims 1 and 8 above. In addition, Nelson discloses:

said condition management structure (Interrupt Source Tree, Col. 3, line 12) interacts only with a logical representation of said microprocessor-accessible registers, node interrelationships, summary bits and masks (Fig. 1A and Col. 3, lines 9-15, and Col. 3, lines 31-44; note that a certain register is a alarm register and an alarm register is a computer device that store summary bits and masks).

16. As for claims 7 and 14, the claims are rejected for the same reasons as claims 1 and 8 above. In addition, Gregory discloses:

said graph generator, said graph converter are embodied in sequences of instructions executable in a general purpose computing system (Figure 7 and Col. 56, lines 52-54).

and Oualline discloses:

said description generator are embodied in sequences of instructions executable in a general purpose computing system (Chapter 17, Section 17.10: Data Structures for a Chess Program, Page 1 of 2).

17. Claims 3, 10, and 15-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Gregory and Nelson in view of Oualline, and further in view of Burke (Sean M. Burke, Perl & LWP, O'Reilly, June 2002).

18. As for claim 15, Gregory discloses:

A computer-implemented system for automatically generating a hierarchical register consolidation structure (A computer-aided circuit analysis tool, Col. 56, lines 52-54), comprising:

a graph generator that parses a High-level Design Language (HDL) file to generate an intermediate graph (Figures 4 and 9, and Col. 27, line 61; note that the tree in Figure 9 is considered as an intermediate graph);

a graph converter, associated with said graph generator, that selectively adds virtual elements and nodes to said intermediate graph (4135, Figure 7) to transform said intermediate graph into a mathematical tree (Figures 7 and 12; the tree in Figure 12 is considered as a mathematical tree);

But Gregory does not explicitly disclose:

an intermediate graph containing definitions of microprocessor-accessible registers, node interrelationships and summary bits, bit offsets and masks associated with alarm registers;

a description generator, associated with said graph converter, that employs said mathematical tree to generate a static tree description in a programming language suitable for use by a device-independent condition management structure and an HTML traversable tree representation based on said mathematical tree.

However, Nelson discloses:

an intermediate graph containing definitions of microprocessor-accessible registers, node interrelationships and summary bits, bit offsets and masks associated with alarm registers (Fig. 1A and Col. 3, lines 9-15 and lines 31-44; note that a certain register is a alarm register and an alarm register is a computer device that store summary bits and masks);

It would have been obvious to one of ordinary skill in the art at the time of invention was made to combine the teachings of Gregory with the teachings of Nelson by making an intermediate graph to contain definitions of microprocessor-accessible registers, node interrelationships and summary bits and masks associated with alarm registers in order to determine the interrupting device (Nelson, Col. 1, line 51).

But neither Gregory nor Nelson explicitly disclose:

a description generator, associated with said graph converter, that employs said mathematical tree to generate a static tree description in a programming language suitable for use by a device-independent condition management structure and an HTML traversable tree representation based on said mathematical tree.

However, Oualline discloses:

a description generator, associated with said graph converter, that employs said mathematical tree to generate a static tree description in a programming language

suitable for use by a device-independent condition management structure (Chapter 17, Section 17.10: Data Structures for a Chess Program, Page 1 of 2).

It would have been obvious to one of ordinary skill in the art at the time of invention was made to combine the teachings of Gregory and Nelson with the teachings of Oualline by employing said mathematical tree to generate a static tree description in a programming language suitable for use by a device-independent condition management structure in order to organize and process information (Oualline, Chapter 1, Page 1 of 4, 1st paragraph).

But none of Gregory, Nelson, and Oualline explicitly discloses:

an HTML traversable tree representation based on said mathematical tree.

However, Burke discloses:

an HTML traversable tree representation based on said mathematical tree (Chapter 9, Example 9-1 and Figure 9-1).

It would have been obvious to one of ordinary skill in the art at the time of invention was made to combine the teachings of Gregory, Nelson, and Oualline with the teachings of Burke by creating an HTML traversable tree representation based on said mathematical tree in order to construct trees from HTML and to process those trees to extract information (Burke, Chapter 9, Page 1 of 2, 2nd paragraph).

19. As for claims 3 and 10, the claims are rejected for the same reasons as claims1, 8, and 15 above. In addition, Burke discloses:

said description generator further generates an HTML traversable tree representation based on said mathematical tree (Chapter 9, Example 9-1 and Figure 9-1).

20. As for claim 16, the claim is rejected for the same reasons as claim 15 above. In addition, Oualline discloses:

said programming language is C (Chapter 17, Section 17.10: Data Structures for a Chess Program, Page 1 of 2).

21. As for claim 17, Gregory discloses:

said HDL file is produced by a hardware description tool (Col. 5, lines 19-22; note that VHDL is a hardware description tool).

22. As for claim 18, the claim is rejected for the same reasons as claim 15 above. In addition, Nelson discloses:

said condition management structure (Interrupt Source Tree, Col. 3, line 12) interacts only with a logical representation of said microprocessor-accessible registers, node interrelationships, summary bits and masks (Fig. 1A and Col. 3, lines 9-15, and

Col. 3, lines 31-44; note that a certain register is a alarm register and an alarm register is a computer device that store summary bits and masks).

23. As for claim 19, the claim is rejected for the same reasons as claim 15 above. In addition, Gregory discloses:

said graph generator, said description generator are embodied in sequences of instructions executable in a general purpose computing system (Figure 7 and Col. 56, lines 52-54).

and Oualline discloses:

said description generator are embodied in sequences of instructions executable in a general purpose computing system (Chapter 17, Section 17.10: Data Structures for a Chess Program, Page 1 of 2).

24. As for claim 20, the claim is rejected for the same reasons as claim 15 above. In addition, Nelson discloses:

said hierarchical register consolidation structure (Interrupt Source Tree, Col. 3, line 12) pertains to a real-time system (Col. 1, lines 9-11).

Response to Arguments

16. Regarding Applicant's remark I with respect to the 101 rejections (second paragraph of Page 6), the same deficiencies remain for claims 1-7 and 15-20 (see the above 101 rejection).

25. Regarding Applicant's remark II (A) (last paragraph of Page 6), Applicant argued that neither Nelson nor Gregory teach or suggest

parsing a High-level Design Language (HDL) file to generate an intermediate graph containing definitions of microprocessor-accessible registers, node interrelationships and summary bits and masks associated with alarm registers

because Nelson does not teach

an intermediate graph (IST) was generated by parsing the HDL file.

Examiner respectfully traverses Applicant's argument. It is true that Nelson does not teach

an intermediate graph (IST) was generated by parsing the HDL file.

However, Gregory teaches:

an intermediate graph was generated by parsing the HDL file (Figures 4 and 9, and Col. 27, line 61; note that the tree in Figure 9 is considered as an intermediate graph);

and Nelson teaches:

an intermediate graph containing definitions of microprocessor-accessible registers, node interrelationships and summary bits and masks associated with alarm registers (Fig. 1A and Col. 3, lines 9-15 and lines 31-44; note that a certain register is a alarm register and an alarm register is a computer device that store summary bits and masks).

It would have been obvious to one of ordinary skill in the art at the time of invention was made to combine the teachings of Gregory with the teachings of Nelson by making an intermediate graph to contain definitions of microprocessor-accessible registers, node interrelationships and summary bits and masks associated with alarm registers in order to provide an interrupt dispatching mechanism which abstracts the interrupting dispatching function from the device drivers and to determine the interrupting device (Nelson, Col. 1, line 51 and Col. 2, lines 24-30).

It is clearly that combination of Gregory's teachings and Nelson's teachings teaches:

parsing a High-level Design Language (HDL) file to generate an intermediate graph containing definitions of microprocessor-accessible registers, node interrelationships and summary bits and masks associated with alarm registers.

Also, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413,

208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

26. Regarding Applicant's remark II (B) (last paragraph of Page 7), Applicant argued that it is not obvious that combination of Gregory's teachings and Nelson's teachings teach:

parsing a High-level Design Language (HDL) file to generate an intermediate graph containing definitions of microprocessor-accessible registers, node interrelationships and summary bits and masks associated with alarm registers

because Nelson does not provide an enabling discloser for parsing a HDL file to generate an intermediate graph containing definitions of microprocessor-accessible registers, node interrelationships and summary bits and masks associated with alarm registers.

Examiner respectfully traverses Applicant's argument. It is true that Nelson does not provide an enabling discloser for parsing a HDL file to generate an intermediate graph. However, Gregory provides an enabling discloser for parsing a HDL file to generate an intermediate graph (Figures 4 and 9, and Col. 27, line 61; note that the tree in Figure 9 is considered as an intermediate graph). As discussed above in response to Applicant's remark II (A), it is obvious that combination of Gregory's teachings and Nelson's teachings teaches:

parsing a High-level Design Language (HDL) file to generate an intermediate graph containing definitions of microprocessor-accessible registers, node interrelationships and summary bits and masks associated with alarm registers because Nelson does not provide an enabling discloser for parsing a HDL file to generate an intermediate graph containing definitions of microprocessor-accessible registers, node interrelationships and summary bits and masks associated with alarm registers.

Also, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.,* 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

27. Regarding Applicant's remark II (C) (last paragraph of Page 8), Applicant argued that combination of Gregory's teachings, Nelson's teachings, and Oualline's teachings does not provide a prima facie case of obviousness of:

parsing a High-level Design Language (HDL) file to generate an intermediate graph containing definitions of microprocessor-accessible registers, node interrelationships and summary bits and masks associated with alarm registers

because Oualline does not provide an enabling discloser for parsing a HDL file to generate an intermediate graph containing definitions of microprocessor-accessible

registers, node interrelationships and summary bits and masks associated with alarm registers.

Examiner respectfully traverses Applicant's argument. It is true that Oualline does not provide an enabling discloser for parsing a HDL file to generate an intermediate graph. However, Gregory provides an enabling discloser for parsing a HDL file to generate an intermediate graph (Figures 4 and 9, and Col. 27, line 61; note that the tree in Figure 9 is considered as an intermediate graph). As discussed above in response to Applicant's remark II (A) and (B), it is clearly that combination of Gregory's teachings and Nelson's teachings provide a prima facie case of obviousness of:

parsing a High-level Design Language (HDL) file to generate an intermediate graph containing definitions of microprocessor-accessible registers, node interrelationships and summary bits and masks associated with alarm registers.

Also, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

28. Regarding Applicant's remark III (Page 9), Examiner respectfully traverses Applicant's argument for the same reasons as described in response to Applicant's remark II.

Conclusion

No new ground of rejection is introduced in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Songwei Qian whose telephone number is 571-270-1910. The examiner can normally be reached on M-F (alternative Friday off 8:00am thru 5:00pm).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wei Zhen can be reached on 571-272-3708. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

SQ
10/05/2007

WEI ZHEN
SUPERVISORY PATENT EXAMINER